Volume 6, No. 2, February 2015
Journal of Global Research in Computer Science
COMMENTARY PAPER
http://rroij.com/global-research-in-computer-science.php
ISSN-2229-371X

SOFTWARE SECURITY

A.Divya
Department of Pharmaceutics, Ratnam Institute of Pharmacy, Nellore, India
Divyapharma15@gmail.com

INTRODUCTION

Security testing helps in securing application. It's a complicated version of software testing. Software Testing mainly focuses on testing of software's practicality. Functions enforced in package area unit analyzed to ensure whether software system produces calculable response. Software testing resembles system's functional aspect. Security testing a lot of advanced than software testing because it considers security, a non-functional system's property. It depicts system's ability to form it secure. The system is made secure by implementing functions which prevent an unauthorized user to access system's valuable and confidential information [1-5]. The developer must code security imposing functions to shield system by preventing it from being exploited. System will be secured if it is well functioned, even in presence of vulnerable code or activity which will exploit system, and does not have any effect on it [6-11].

Providing security to system is incredibly advanced as compared to easy software package testing method that involves recording machine and white box testing. For securing system, we want to ascertain system's two important things: First, validity of enforced security measures that provide functionality and security to system [12-17]. Security measures conjointly include features like cryptography, robust authentication, and management measures. And second, system's behavior when it gets attacked by attackers, leading to destruction by accessing secured and confidential guidance. Attackers will prohibit attacker's exploitable activities for hacking system.

Security testing is incredibly important for software application because it takes care of confidential information. It ensures that confidential information does not get unnoticed by unauthorized entity. It works on the farside purposeful (i.e. black box) and implementation (i.e. white box) testing. Security tester might use several techniques to locate system's vulnerabilities. Testing system's security checks the loopholes or vulnerabilities in system which may cause failure of security functions of system ultimately resulting to great losses to organization [18]. Therefore, security testing is used to confirm that developed software is free from flaws and hence, the system is safe from unauthorized individual, be it an employee or an outsider.

Security testing identifies threats and calculates its result on system. The impact is analyzed by developers or testers. They place their efforts to interrupt the system or to urge into it to find bugs. So, security testing is extremely essential in IT sector for information protection. Security testing is said to risk primarily based testing approach that analyzes risk in each stage. Proper measures are taken to eliminate risk to create system secure. So, Testers should incorporate a risk-based testing approach by keeping system's subject area reality and attacker's mindset into thought for applying software security adequately [19-22]. During this approach, risk affected areas are known for testing. Developers/ Tester need to develop check cases to reveal issues if any. The approach provides high level of software security as compared to black-box testing. Security testing deals with system's security. It observes system's behavior in presence of malicious attack. It tries to construct and execute test cases to create code work properly in attack part as well [23-28].

Software security is explained through varied perspectives, a user needs his system to figure and to way the developers ought to use their inventive minds towards making a secured software system.

REFERENCES

[1] G.Tian-yang, S.Yin-sheng, and F.You-yuan. "Research on Software Security Testing", World Academy of Science, Engineering and Technology, Vol. 70, p 647-651, 2010.

[2] G. Hoglund and G. McGraw. Exploring Software: How to Break Code, Addison-Wesley, 2004.

[3] G.Erdogan, K.Stolen, "Risk-driven Security Testing versus Test-driven Security Risk Analysis", First Doctoral Symposium on Engineering Secure Software and Systems.

[4] K.K. Aggarwal, Y.Singh, A.Kaur and R.Malhotra. "Software Engineering", (3rd ed.), Copyright © New Age International Publishers, 2005.

[5] ME Khan & F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques", International Journal of Advanced Computer Science and Applications, (IJACSA) Vol. 3, No.6, 2012.

[6] Jovanovic, Irena, "Software Testing Methods and Techniques".

[7] Ould, M. A. Managing software quality and business risk. Chichester: John Wiley & Sons, 1999.

[8] T. DeMarcoand, T. Lister. Waltzing with Bears: Managing Risk on Software Projects, 2003.

[9] B. Potter and G. McGraw, "Software Security Testing", IEEE Security & Privacy, pp. 32-36, 2004.

[10]SA Khan, RA Khan "Software Security Testing Process",

Proc. of the Intl. Conf. on Recent Trends In Computing and Communication Engineering-- RTCCE, p39-42, 2013.

[11]H.H Thompson. “Why security testing is hard”, IEEE Security & Privacy, v 1, n 4, pp. 83-6, 2003.

[12]B. Potter, G. Mcgraw. “Software Security Testing,” IEEE Security & Privacy, v2, n5, pp. 81-85, 2004.

[13]S. Barnum, G. Mcgraw. “Knowledge for Software Security”, IEEE Security & Privacy, v3, n2, pp. 74-78, 2005.

[14] J. Viega and G. McGraw. Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley, 2001.

[15]G. McGraw, “Software Security: Building Security in”, Addison-Wesley Professional, 2006.

[16]Online Documentation, August 2013. URL: http://www.guru99.com/what-is-security-testing.html.

[17]G. McGraw, “Testing for Security During Development: Why We Should Scrap Penetratre-and-Patch,” IEEE Aerospace and Electronic Systems, Vol. 13, no. 4, pp. 13-15, 1998.

[18]K.El Emam. “A Methodology for Validating Software Product Metrics”, Ottawa, Ontario, Canada, National Research Council of Canada, 2000.

[19]IK. El-Far. and AJ. Whittaker. “Model based Software Testing” Encyclopedia on Software Engineering, Wiley, pp. 1-22, 2001.

[20]C. Kener, Senior Member, IEEE, and W P. Bond, “Software Engineering Metrics: What Do They Measure and How Do We Know?” 10th International Software Metrics Symposium, 2004.

[21]FS. Li, W-M. Ma, C. Architect, “Architecture Centric Approach to Enhance Software Testing Management”, Eighth International Conference on intelligent Systems Design and Applications, pp. 654-659.

[22]L. Yongzhong, S. Simeng Da, J. Jun Yang, N. Songlin, “Research on a Behavior-Based Software Test Process Model”, International Symposium on Computer Science and Computational Technology, pp. 114- 117. 2008.

[23]N. Nagappan, L.Williams, M. Vouk, Osborne, “Early Estimation of Software Quality Using In-Process Testing Metrics: A Controlled Case Study”, Third Software Quality Workshop, pp. 46-52, 2005.

[24]R A. Paul, T.L. Kunii, Y. Shinagawa, and MF. Khan. “Software Metrics Knowledge and Databases for Project Management” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.11, NO.1, pp. 255-264, 1999.

[25] T. Repasi, “Software Testing- State of the Art and Current Research Challenges” 5th International Symposium on Applied Computational Intelligence and informatics, pp. 47-50, 2009.

[26]NF. Schneidewind. “Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics”, IEEE Explore, pp. 232-239, 1989.

[27]W. Afzal and R. Torkar, “Incorporating Metrics in an Organizational Test Strategy” IEEE International Conference on Software Testing Verification and Validation Workshop, 2008.

[28]K. K Aggarwal and Y. Singh, “Software Engineering”, New Age International, New Delhi, 2005.